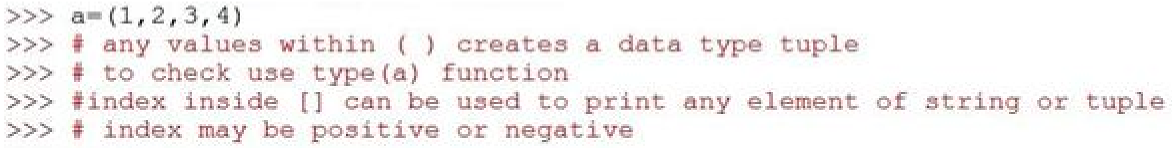
Tuple

Introduction



```
>>> a=(1,2,3,4)
>>> # any values within ( ) creates a data type tuple
>>> # to check use type(a) function
>>> #index inside [] can be used to print any element of string or tuple
>>> # index may be positive or negative
```

Suppose it is mandatory to have the following types of food in the lunch menu of a Restaurant. **Welcome Drink, Veg Starter, Non-Veg Starter, Veg Main Course, Non-Veg Main Course, Dessert**

How can we store it such that no one can modify the elements?

This is where we can use a data type known as **tuple**.

Tuple can store a sequence of elements but the value of the elements cannot be changed. (i.e. tuples are **IMMUTABLE**). Elements can be homogeneous or heterogeneous but they are **READ-ONLY**.

Creating a tuple	lunch_menu=("Welcome Drink","Veg Starter","Non-Veg Starter","Veg Main Course","Non-Veg Main Course","Dessert")	() are optional, a set of values separated by comma is also considered to be a tuple. sample_tuple="A","B","C" Although () are optional, it is a good practice to have them for readability of code. If we need to create a tuple with a single element, then we need to include a comma as shown below: sample_tuple=("A",)
Random Write	lunch_menu[0]=""	This will result in an error as tuple is immutable. Hence random write is not possible in tuple.

P1. Which among the following statements may result in an error? Assume that the statements are executed in the order in which it is written.

a. tup1=(5,10,15,20,25)
b. print(len(tup1))
c. print(tup1[4])
d. print(tup1[5])
e. print(tup1[4:5])

f. tup1[2]=12
g. print(tup1)
h. tup1=tup1+(8,9)

P2. Pure Gems Store sells different varieties of gems to its customers. **Emerald, Ivory, Jasper, Ruby, Garnet** and their prices are **1760, 2119, 1599, 3920, 3999** respectively.

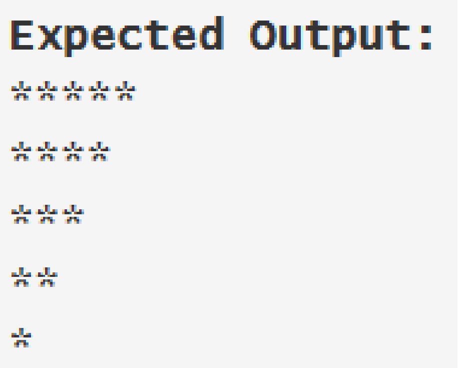
Write a Python program to calculate the bill amount to be paid by a customer based on the list of gems and quantity purchased. Any purchase with a total bill amount above Rs.30000 is entitled for 5% discount. If any gem required by the customer is not available in the store, then consider total bill amount to be -1.

Assume that quantity required by the customer for any gem will always be greater than 0.

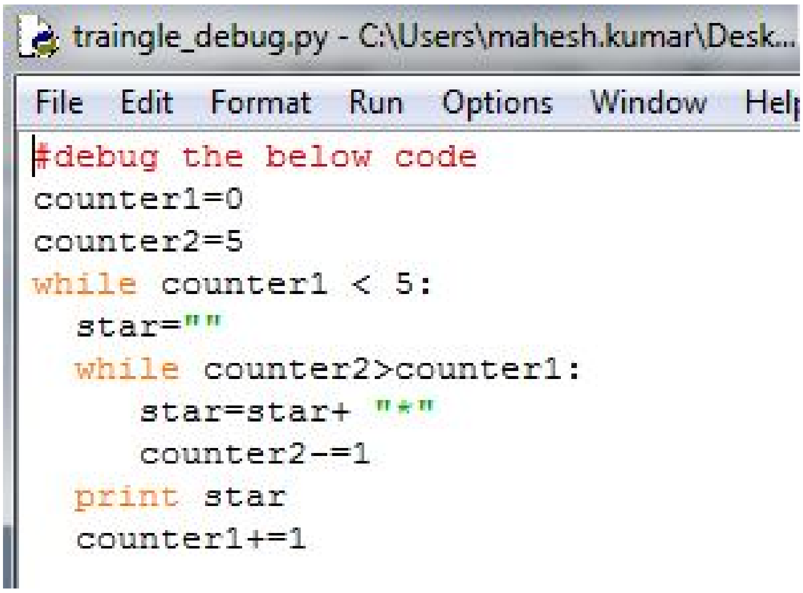
Perform case-sensitive comparison wherever applicable.

P3. Write a python function to check whether three given numbers can form the sides of a triangle.
Hint: Three numbers can be the sides of a triangle if none of the numbers are greater than or equal to the sum of the other two numbers.

P4. Execute the program **triangle.py** available in the folder. The program **triangle.py** is written to display "*" as per the expected output given below. But the code is having logical errors, debug the program and correct it.



```
Expected Output:
*****
****
***
**
*
```


traingle_debug.py - C:\Users\mahesh.kumar\Desk...
File Edit Format Run Options Window Help

```
#debug the below code
counter1=0
counter2=5
while counter1 < 5:
   star=""
   while counter2>counter1:
      star=star+ "*"
      counter2-=1
   print star
   counter1+=1
```

P5. Write a python program to solve a classic ancient Chinese puzzle.

We count 35 heads and 94 legs among the chickens and rabbits in a farm. How many rabbits and how many chickens do we have?

Sample Input	**Expected Output**
heads-150 legs-400	100 50
heads-3 legs-11	No solution
heads-3 legs-12	0 3
heads-5 legs-10	5 0